

WHAT IS CLAIMED IS:

1. A computer-implemented method for a client to interact with a server, the computer-implemented method comprising:
 - creating a cached object from an original object, the original object being managed by the server; and
 - establishing a notification bond with the server, the notification bond enabling the client to obtain a notification from the server in response to an object related event associated with the original object.
2. The computer-implemented method of Claim 1, wherein the object related event includes when the original object has been modified.
3. The computer-implemented method of Claim 1, wherein establishing the notification bond is performed in response to creating the cached object.
4. The computer-implemented method of Claim 1, further comprising:
 - obtaining a notification from the server; and
 - updating the cached object using the notification.
5. The computer-implemented method of Claim 1, wherein obtaining the notification includes retrieving a notification log containing the notification.
6. The computer-implemented method of Claim 1, further comprising:
 - reconnecting with the server after a disconnected period of time;
 - requesting a notification log containing a notification; and
 - synchronizing the cache object with the original object using the notification.
7. The computer-implemented method of Claim 1, further comprising maintaining states associated with the notification bond.

8. The computer-implemented method of Claim 7, wherein the states are maintained in a persistent medium.

9. The computer-implemented method of Claim 8, wherein the states include a bond number that uniquely identifies the notification bond.

10. The computer-implemented method of Claim 8, wherein the states include a aggregate bond number that is unique to the client.

11. The computer-implemented method of Claim 8, further comprising reestablishing the states on the client after a restart.

12. The computer-implemented method of Claim 11, further comprising synchronizing the states on the client and corresponding states on the server.

13. A computer-implemented method for a server to interact with a client, the computer-implemented method comprising:

establishing a notification bond with the client, the notification bond enabling the client to obtain a notification from the server in response to an object related event associated with an object; and

enabling the client to cache the object.

14. The computer-implemented method of Claim 13, wherein the object related event includes when the object has been modified.

15. The computer-implemented method of Claim 13, wherein establishing the notification bond is performed in response to a request from the client to cache the object.

16. The computer-implemented method of Claim 13, further comprising: determining an object related event that was not caused by the client;

creating a notification in accordance with the notification bond; and
providing the notification to the client.

17. The computer-implemented method of Claim 13, further comprising:
determining an object related event that was not caused by the client;
creating a notification in accordance with the notification bond; and
recording the notification in a notification log.

18. The computer-implemented method of Claim 17, further comprising:
establishing a connection with the client; and
sending the notification log to the client.

19. A distributed file system for sharing objects, comprising:
a server configured to manage original objects, the server including a
bond manager configured to issue notification bonds to clients, each notification bond
enabling a client to obtain a notification from the server in response to an object related
event associated with an original object.

20. The distributed file system of Claim 19, wherein the bond manager is
configured to provide notifications to the clients in accordance with the notification
bonds.

21. The distributed file system of Claim 19, wherein the server further
comprises a file system manager configured to manage the original objects and wherein
the bond manager comprises a filter component configured to determine object related
events by monitoring communication traffic associated with the file system manager.

22. The distributed file system of Claim 19, wherein the bond manager is
configured to maintain a bond table and wherein the bond table includes states that
relate each notification bond with an original object and a client to whom the
notification is to be provided.

23. The distributed file system of Claim 19, wherein the bond manager is configured to maintain a notification log and wherein the notification log includes notifications for the client.

24. The distributed file system of Claim 19, further comprising:
a client configured to create a cached object associated with an original object managed by the server, the client including a notification handler configured to maintain a notification bond associated with the original object in conjunction with the server.

25. The distributed file system of Claim 19, wherein the notification handler is configured to obtain from the server a notification log associated with the notification bond and to update the cached object in accordance with the notification log.

26. The distributed file system of Claim 25, wherein the notification log includes notifications associated with a plurality of notification bonds.

27. The distributed file system of Claim 19, wherein the notification handler is configured to maintain a bond table and wherein the bond table includes states that relate each notification bond with a cached object and a server that manages an original object corresponding to the cached object.

28. A computer-readable medium encoded with a data structure, comprising:
a first indexing data field containing object identifiers, each object identifier uniquely identifying an object that is managed by a server; and
a second data field containing entries, each entry being indexed to an object identifier in the first indexing data field and containing states associated with a notification bond between the server and a client that caches the object identified by the object identifier.

29. The computer-readable medium of Claim 28, wherein each object identifier in the first indexing data field includes a file path associated with an object.

30. The computer-readable medium of Claim 29, wherein each object identifier includes a hash of the file path.

31. The computer-readable medium of Claim 28, wherein each entry of the second data field includes a bond number that uniquely identifies a notification bond.

32. The computer-readable medium of Claim 28, wherein each entry of the second data field includes a client identifier that identifies a client associated with a notification bond.

33. The computer-readable medium of Claim 28, wherein each entry of the second data field includes a type identifier that identifies a type associated with a notification bond.

34. A computer-readable medium encoded with a data structure, comprising:
a first indexing data field containing server identifiers, each server identifier uniquely identifying an server that manages an original object, the original object being cached by a client; and
a second data field containing entries, each entry being indexed to an server identifier in the first indexing data field and containing states associated with a notification bond between the client and a server identified by the server identifier, the notification bond being associated with a cached object created by the client from an original object.

35. The computer-readable medium of Claim 34, wherein each entry of the second data field includes a bond number that uniquely identifies a notification bond.

36. The computer-readable medium of Claim 34, wherein each entry of the second data field includes an original object identifier that identifies an original object.

37. The computer-readable medium of Claim 34, wherein each entry of the second data field includes a cached object identifier that identifies a cached object associated with an original object.

38. The computer-readable medium of Claim 34, further comprising a third data field that includes an aggregated bond number.

39. The computer-readable medium of Claim 34, further comprising a third data field that includes a notification log offset.

40. A distributed file system for sharing objects, comprising:
means for a client to cache an original object managed by a server; and
means for establishing a notification bond with the server and the client, the notification bond enabling the client to obtain a notification from the server in response to an object related event associated with the original object.

41. The distributed file system of Claim 40, further comprising:
means for obtaining a notification from the server; and
means for updating the cached object using the notification.

42. The distributed file system of Claim 40, further comprising:
means for reconnecting with the server after a disconnected period of time;
means for requesting a notification log containing a notification; and
means for synchronizing the cache object using the notification.

43. The distributed file system of Claim 40, further comprising:
means for determining an object related event;

means for creating a notification in accordance with the notification bond; and

means for providing the notification to the client.

44. The distributed file system of Claim 40, further comprising:
means for determining an object related event;
means for creating a notification in accordance with the notification bond; and

means for recording the notification in a notification log.

45. The distributed file system of Claim 40, further comprising:
means for establishing a connection with the client; and
means for sending the client the notification log.

46. The distributed file system of Claim 40, further comprising:
means for the client to drop the notification bond.

47. The distributed file system of Claim 40, further comprising:
means for the server to drop the notification bond.

48. The distributed file system of Claim 40, further comprising:
means for the server to drop all notification bonds associated with the client.

49. The distributed file system of Claim 48, further comprising:
means for the server to reset clear states associated with the dropped notification bonds.

50. A computer-implemented method for maintaining cached objects that correspond to original objects managed by a server, the computer-implemented method comprising:

reestablishing a communication link between the client and the server after a period of time without a communication link;

obtaining notifications from the server about changes made to at least one of the original objects during the period of time; and

synchronizing a cache object corresponding to the at least one original object using the notifications, without synchronizing all of the cached objects.

51. A computer-implemented method for synchronizing cached objects maintained by a client with the corresponding original objects maintained by a server, the computer-implemented method comprising:

persistently maintaining, by the server, server bond states related to the original objects, the server bond states corresponding to notification bonds associated with the original objects, each notification bond enabling the client to obtain a notification from the server when at least one of the original object has been modified; and

persistently maintaining, by the client, client bond states corresponding to the server bond states.

52. The computer-implemented method of claim 51, further comprising reestablishing the server bond states after a server reboot or restart.

53. The computer-implemented method of claim 52, further comprising recovering notifications associated with the notification bonds after a server reboot or restart.

54. The computer-implemented method of claim 51, further comprising reestablishing the client bond states after a client reboot or restart.

55. The computer-implemented method of claim 54, further comprising recovering notifications associated with the notification bonds after a server reboot or restart.

56. The computer-implemented method of claim 51, further comprising determining, by the server, to drop a notification bond; performing, by the server, an operation to drop the notification bond; and providing, by the server, a notification to the client for dropping the notification bond.

57. The computer-implemented method of claim 56, wherein performing the operation commits the server to dropping the notification bond.

58. The computer-implemented method of claim 57, wherein performing the operation is completed before providing the notification to the client.

59. The computer-implemented method of claim 51, further comprising determining, by the server, to drop all notification bonds associated with the client;
performing, by the server, an operation to drop the notification bonds;
and
providing, by the server, a notification to the client for dropping the notification bonds.

60. The computer-implemented method of claim 59, wherein performing the operation commits the server to dropping the notification bonds.

61. The computer-implemented method of claim 60, wherein performing the operation is completed before providing the notification to the client.

62. The computer-implemented method of claim 51, further comprising determining, by the client, to drop a notification bond; performing, by the client, an operation to drop the notification bond; and requesting the server to drop the notification bond.

63. The computer-implemented method of claim 62, wherein performing the operation commits the client to dropping the notification bond.

64. The computer-implemented method of claim 63, wherein performing the operation is completed before requesting the server to drop the notification bond.